REMARKS

In the specification, paragraphs **[0024]**, **[0025]**, **[0042]**, **[0046]** and **[0048]** have been amended to correct minor errors which were not detected prior to filing.

In the drawings, Figures 2 and 3 have been amended to add previously omitted reference numerals, shown on annotated sheets submitted herewith.

The amendments do not involve the introduction of new matter.

Applicant's note with appreciation that the Examiner has allowed claims 17-28 and has acknowledged that claims 6, 9-11 and 13-15 are directed to allowable subject matter.

Claim Rejections - 35 U.S.C. §103(a)

Claims 1-5, 7-8 and 12 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over primary reference ***Imada*** U.S. Patent No. 4,797,886 in view of secondary reference ***Ozawa*** U.S. Patent No. 6,038,649. Applicants have carefully reviewed the cited references and believe that they neither independently nor in combination, teach or remotely suggest the invention of the rejected claims. Accordingly, favorable reconsideration is respectfully requested in light of the following comments.

The Examiner admits that neither reference individually teaches the features of the claims under rejection. The Examiner has not indicated specifically the features which would be extracted from secondary reference ***Ozawa*** and incorporated into primary reference ***Imada***, how that modification would be accomplished, specifically what desirability or motivation there is in either reference to do so, or shown that the resulting combination would be the same as the invention claimed in the rejected claims.

The Examiner is reminded at the outset that in order to establish a prima facie case of obviousness, three basic criteria must be met as stated at MPEP § 2143:

> "First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine

> reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.
>
> The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)".

The Examiner has not explicitly indicated how the references would be combined or the specific suggestion or motivation in the references to make the modification.

The Present Invention

As explained in the background of the present specification, memory BIST controllers use a very wide instruction word (e.g., 40 bits) for programming algorithms. A memory test developer may specify as many instructions as required in a memory BIST microprogram memory array to perform a memory test. However, since the developer wishes to keep the number of gates required to implement a word in the memory array to a minimum, it is also desirable to keep the number of instructions to a minimum. Generally, each word requires about 250 gates.

Many memory test algorithms, particularly March algorithms, perform the same operations repeatedly but with opposite data or parameters or traverse an address space in one direction and then in the opposite direction. The conventional way of doing this utilizes many more instructions than are required which, in turn, requires many more gates than required.

Thus, the present invention seeks to provide a method for repeating memory test instructions in a simple and efficient manner so as to keep the number of instructions to a minimum while allowing the use of a very wide instruction word.

Applicants have found that this objective can be achieved by providing each test instruction with a "repeat control field". The repeat control field is provided with an active value when the instruction is the last instruction of a group of instructions to be repeated. The remaining instructions are provided with an inactive value. In the method of the present invention, instructions in a memory array are executed in sequence until an instruction that has an active repeat control field (last instruction) is encountered. After completing the last instruction, the instructions of the group of instructions associated with the last instruction, are executed in sequence starting from a group start instruction. Thus, the method does not require a large register for storing a lengthy last or end instruction.

After executing a repeat cycle of the group of instructions, predetermined fields of the instructions in the group are modified in accordance with predetermined field modification instructions. This obviates the need for additional instructions having the modified instructions, as is required in conventional methods.

Independent claims 1 and 7 are directed to a method of testing memory which includes executing test instructions having "repeat control fields". These method steps are reflected in independent claim 1 as follows:

> "executing each instruction of a plurality of test instructions in sequence, each said instruction having an inactive repeat control field <u>except for a last instruction of each of one or more groups of one or more instructions to be repeated</u>, each said last instruction having an active repeat control field;
>
> <u>for each instruction having an active repeat control field: executing in sequence the instructions of the group of instructions with which said each instruction is associated for a predetermined number of repeat cycles for said group</u>; and
>
> <u>for each said repeat cycle, modifying predetermined fields of each instruction in accordance with a predetermined field modification instructions for each said repeat cycle</u>." (emphasis added)

The Prior Art

Applicants have carefully reviewed the cited references and cannot find any teaching or suggestion of performing a repeat procedure by incorporating a repeat control field into test program instructions. In general, Imada is not concerned with and does not discuss repeating a group of instructions in a program memory array. Imada does disclose repeating a single instruction with different memory addresses generated by an internal address generator, while the program counter is essentially disabled. Ozawa discloses a method of repeating program instructions, but does so in a manner which does not involve the use of a repeat control field in the program instructions and does not modify predetermined fields of the group of instructions being repeated. Even with the benefit of hindsight, which is improper, Applicants fail to appreciate how the teachings of the references can be combined to provide the claimed invention.

IMADA

Imada has been cited as the primary reference. The Examiner admits (page 3, last line of office action) that ***Imada*** does not disclose instructions having repeat control fields.

Imada is concerned with the problem of synchronizing a tester with a memory that has its own embedded address generator. This is a very different problem from that with which the present Applicant's are concerned. More specifically, Imada seeks to provide a memory test pattern generator for generating test patterns for a special type of memory called "nibble memory". Nibble memory has an internal address generating means and is capable of address accessing by merely supplying a clock signal. Imada describes the problem with which it is concerned at col. 3, lines 34 onward:

> "In a tester, the same addresses as those internally generated in the DUT 13 have to be generated because it is necessary to identify the address of the DUT 13, from which data is being read out. That is, the same address as that internally generated in the DUT 13 also has to be externally generated for comparison and accessing to the fail memory 15. Heretofore, operation instructions for generating the same addresses as internal addresses are written in the address operation instruction area 1B, and the same addresses as the internal addresses to be accessed are generated in the periods MN, NN and NM of the internal address mode according to the operation instructions."

Clearly, the problem which Imada seeks to overcome is very specific to nibble memory and is not the problem with which the present invention is concerned. Imada neither seeks to solve the problems associated with repeating blocks of program instructions nor describes any method for doing so.

The internal test pattern generator described by Imada stops program counter 2 in response to a predetermined value of code ***IN*** contained in instruction code area 1A of a program instruction. The contents of instructions are described in the paragraph beginning at line 31 of column 4. There is no mention therein, or elsewhere, of a repeat control field, let alone for the purpose described and claimed by Applicants. This is admitted by the Examiner.

Imada discloses a repeat controller 21 which is specifically designed for use with the internal test pattern generator for incrementing an internal address under control of a clock signal. The repeat controller controls the number of times internal address generation is performed while the program counter is stopped and is operative when the memory is placed in internal address mode, indicated by a signal IN=1, as explained in col. 8, lines 5 onward.

The repeat controller of Imada is unique to the internal test pattern generator of Imada, does not perform the functions of the repeat module of the present invention and, therefore, cannot be substituted for the repeat module of the present invention.

OZAWA

Unlike Imada, Ozawa is concerned with repeating a block of instructions. Ozawa seeks to overcome problems the conventional method for generating addresses in circular addressing of memory which requires three operations (addition or subtraction) to generate a next address and, consequently, three operation units are needed for the address generating circuit. Conventional address generating circuits require three adders and, since adders are relatively large hardware units with respect to the circuit scale, the conventional address generating circuit containing three adders with a large circuit scale occupies a large area on the semiconductor chip. This is considered a problem. Also, for calculating address generation, it is necessary to store the step, top address, and bottom address. This is also considered to be a problem.

To overcome these problems, Ozawa provides an address generating circuit 7, shown in Figure 6, for this purpose. The circuit includes a register 76 to hold the end address of the block of instructions and a register 80 to hold the starting address of the block. An instruction address, maintained by a program counter 72, is compared to the end address in register 76. When the end address is detected, the program counter is loaded with the starting address of the block of instructions stored in register 80 to begin repeating the block of instructions. A block repeat count register 86 maintains a repeat count.

A primary difference between the present invention and that of Ozawa is that the present invention modifies the instructions themselves so that the instruction or block of instructions being repeated is slightly different on each repetition because of the field modification instructions. Further, the present invention detects the end of a loop is by monitoring the repeat control field instead of using a comparator to check whether the address in the program counter is equal to a specific end address.

The present invention does require a register for holding the end address or circuitry for comparing the current address with the end address to determine the end of the block. The end address is determined by an instruction having a active repeat control field.

Ozawa also provides a zero detection circuit 70 which delays decrements of repeat count register 86 by a number of clock cycles that is equivalent to a pipeline depth for instruction prefetching of a processor connected to program counter 72. Zero detection circuit 70 outputs a loop-end control signal which controls a selector to selectively provide an incremented address or the start address to the program counter. By delaying decrements of register 86, the state of the repeat count is correctly maintained when the processor pipeline is flushed during an interrupt. The zero detection circuit also deactivates a loop-end control signal for the number of clock cycles equivalent to the depth of the prefetch pipeline during the final repeat loop iteration(s) so that a loop with a block size less than or equal to the depth of the prefetch pipeline can be repeated the correct number of times.

The present invention does not require a zero detection circuit. Clearly, the method employed by Ozawa is different from the method described and claimed by Applicants.

Like Imada, Ozawa neither teaches nor even remotely suggests providing program instructions with a repeat control field to indicate that the current address is the last address of a group of instructions to be repeated. In addition, Applicants are unable to find any motivation in either patent of doing so. Thus, the patents, taken independently or in combination, cannot perform the method of independent method claims 1 and 7 which provide for executing a sequence of instructions until an instruction having an active repeat control field is encountered at which point, the group of instructions associated with that instruction is repeated.

Claims 1-5, 7 and 8

Turning to the Examiner's comments respecting claim 1 on page 3 of the Office Action, Imada does not disclose "executing instructions of a group of instructions with which said each instruction is associated ...". The term "said each instruction" refers to an instruction "having an <u>active</u> repeat control field". The Examiner admits that Imada does not disclose instructions having repeat control fields. Imada also does not disclose modifying instructions "for each repeat cycle". In the present invention, the same modified data is used by each instruction of the group that is being repeated. Imada teaches modifying data for a particular instruction using internal address generator 23 while the circuit is in internal address mode. As already mentioned, the repeat register of Imada merely specifies the number of times an internal address is generated; it does not specify the number of times a "group of instructions" is executed.

The repeat module of the present invention is not used for generating a "releasing signal" as suggested by the Examiner in the fourth paragraph of page 4 of the Office Action. Unlike Imada, the present invention does not suspend operation of a program counter.

With regard to the Examiner's reference to col. 22, lines 1-8 of Ozawa in the first paragraph on page 5 of the Office Action, Applicants note that this text is part of claim 1 of the patent. In any case, the referenced clause refers to the state of a loop-end control signal which is shown in Figure 6 of the patent. The loop-end control signal is not a field in a program instruction and, therefore, is not at all the same as the repeat control field called for in claims 1 and 7.

With reference to the second full paragraph on page 5 of he Office Action, the Examiner suggests that:

> "one would modify Imada's memory testing to explicitly including the repeat execution of the program in memory region as well as the active and inactive control signaling used via program counter as taught by Ozawa's address generating circuit of simple configuration for repeating a selected block of instruction in supporting the compute memory array testing."

The Examiner's statement is not understood because Imada does not discuss repeating instructions loaded in program memory. Accordingly, such a modification of Imada would have to be an addition to the Imada circuitry and would not result in the method and circuit of the present invention. The Examiner has not identified any statement in Imada which would serve as motivation to make such as modification. It is also not clear how the modification would be made. In other words, would one replace repeat controller 21 with circuitry from Ozawa? Imada requires repeat controller 21 to apply signals to program counter control 3, internal address generator 23, data generator 5 and timing set control 26. Clearly, removal of repeat controller 21 would be contrary to the teachings of Imada, which is improper. Alternatively, it is not clear how repeat controller 21 would interact with the circuitry of Ozawa. The Examiner has not described precisely what circuitry would be taken from Ozawa and incorporated into Imada.

With reference to the first full paragraph on page 6 of Office Action, the Examiner is reminded that the motivation must come from the references and not from hindsight after having had the benefit of reading Applicant's specification. Again, the Examiner has not identified any statement in either reference to support the motivation suggested by the Examiner.

Claims 2-5, 7 and 8 depend from either claim 1 or 7 and, accordingly, these claims will distinguish from the applied references for the same and additional reasons. However, Applicant submits the following additional comments.

With regard to claim 2, the Examiner states that the paragraph at col. 5, lines 11-15 discloses "loading field modification commands for each repeat cycle into a field modification register associated with said each repeat cycle." Applicants are unable to find any such teachings in the referenced paragraph. Similarly, there are no teachings with respect to the the loading of the number of repeat cycles to be performed into a repeat cycle register. Reconsideration is respectfully requested.

With regard to the rejection of claims 3-5, Applicant's are unable to find any mention of modifying instruction fields in the referenced portions of Imada. Reconsideration is respectfully requested.

Similarly, with regard to the rejection of claim 8, Applicants are unable to find in col. 8,lines 34-55 of Imada, any teachings of a "next instructions field in a program instruction for containing one or more conditions fields for determining the next instruction to be executed". This field is described on page in paragraph **[0035]** of Applicant's specification. Reconsideration is respectfully requested.

Claim 12

Claim 12 is directed to an improvement in a memory test controller comprising:

> "a repeat module for repeating a group of one or more test instructions with modified data, said repeat module including <u>storage means for storing instruction field modification data</u>; and <u>each register of said test instruction register array including an instruction field for enabling or disabling said repeat module</u>."

The storage means for storing instruction field modification data referred to in claim 12 are registers 61, 62 and 63 shown in Figure 5 of the present application. For each cycle of operation (group or block of instructions), the repeat module selects the contents of one of these registers for use in all of the instructions of the group of instructions to be repeated.

The Examiner asserts that Imada discloses "a repeat module for repeating one or more test instructions" (page 9, third last paragraph of the Office Action). As mentioned earlier, Applicants respectfully submit that Imada does not disclose repeating a group of one or more instructions.

The Examiner also asserts that Imada discloses "storage means for storing instruction field modification data" and relies on the description at col. 9, lines 45-55 for support:

> "As has been described in the foregoing, according to the invention the signal IN representing the internal address mode (IN=1) is stored in the instruction code area 1A, and the data representing the number of times to generate an internal address is stored in the repeat register 22. Thus, the counter 34 can hold the program counter 2 and also put the address and data generators 4 and 5 in the inhibit state. Further, the timing set controller 28 generates the timing data TS0, TS1 and TS2 at the time of the internal address mode. "

Applicants are unable to find any teachings in this paragraph of storing field modification data. The only storage means mentioned is repeat register 22. Register 22 does not store data that will be used to modify data in an instruction. "Instruction code area 1A" is part of a program instruction and the contents of area 1A is not field modification data. The storage means contemplated by Applicants does not involve a field in a program instruction.

The discussion on pages 10 and 11 of the Office Action appears to be substantially the same as pages 4-6 and is believed to have been addressed above.

In view of the foregoing, Applicants believe and respectfully submit that the claims under rejection are patentable of the art of record and that the application is in condition for allowance. Early favorable reconsideration and action to this end is respectfully requested.

Respectfully Submitted,

Eugene E. Proulx
Registration No. 35,815

e-mail: gene@logicvision.com
phone: 613-722-2051 ext. 240

Annotated Sheet Showing Changes

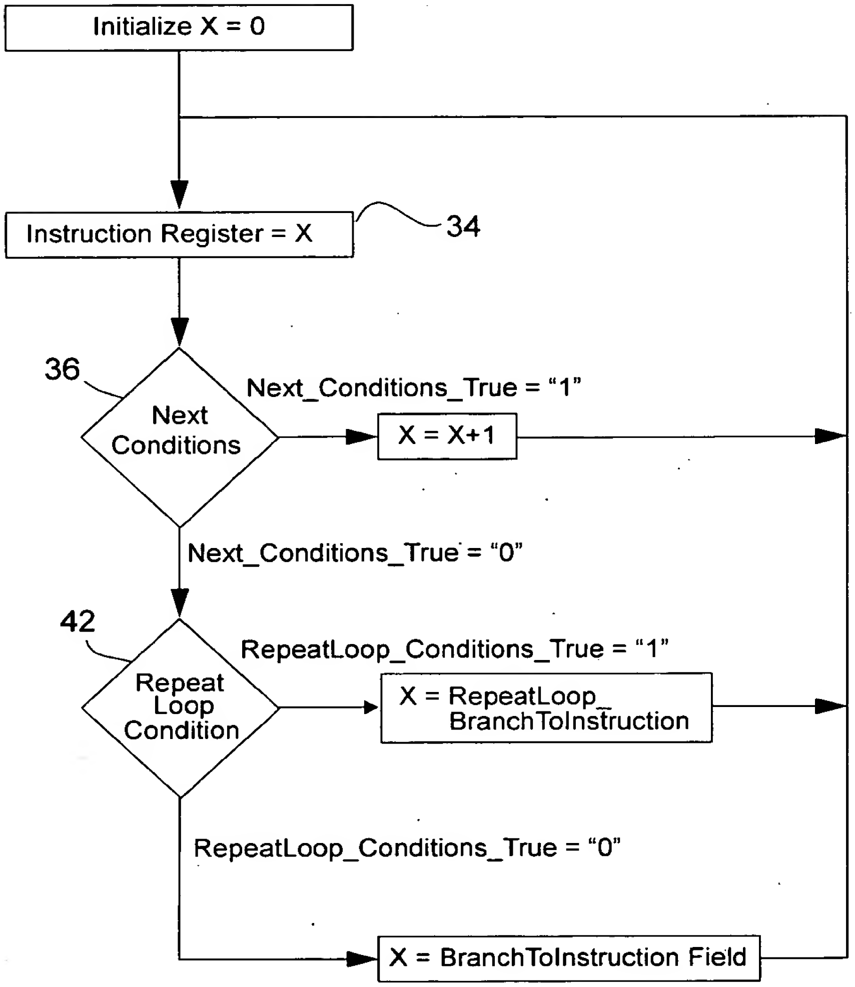
Initialize X = 0
Instruction Register = X
34
36
Next Conditions
Next_Conditions_True = "1"
X = X+1
Next_Conditions_True = "0"
42
Repeat Loop Condition
RepeatLoop_Conditions_True = "1"
X = RepeatLoop_ BranchToInstruction
RepeatLoop_Conditions_True = "0"
X = BranchToInstruction Field

Fig. 2.

Annotated Sheet Showing Changes

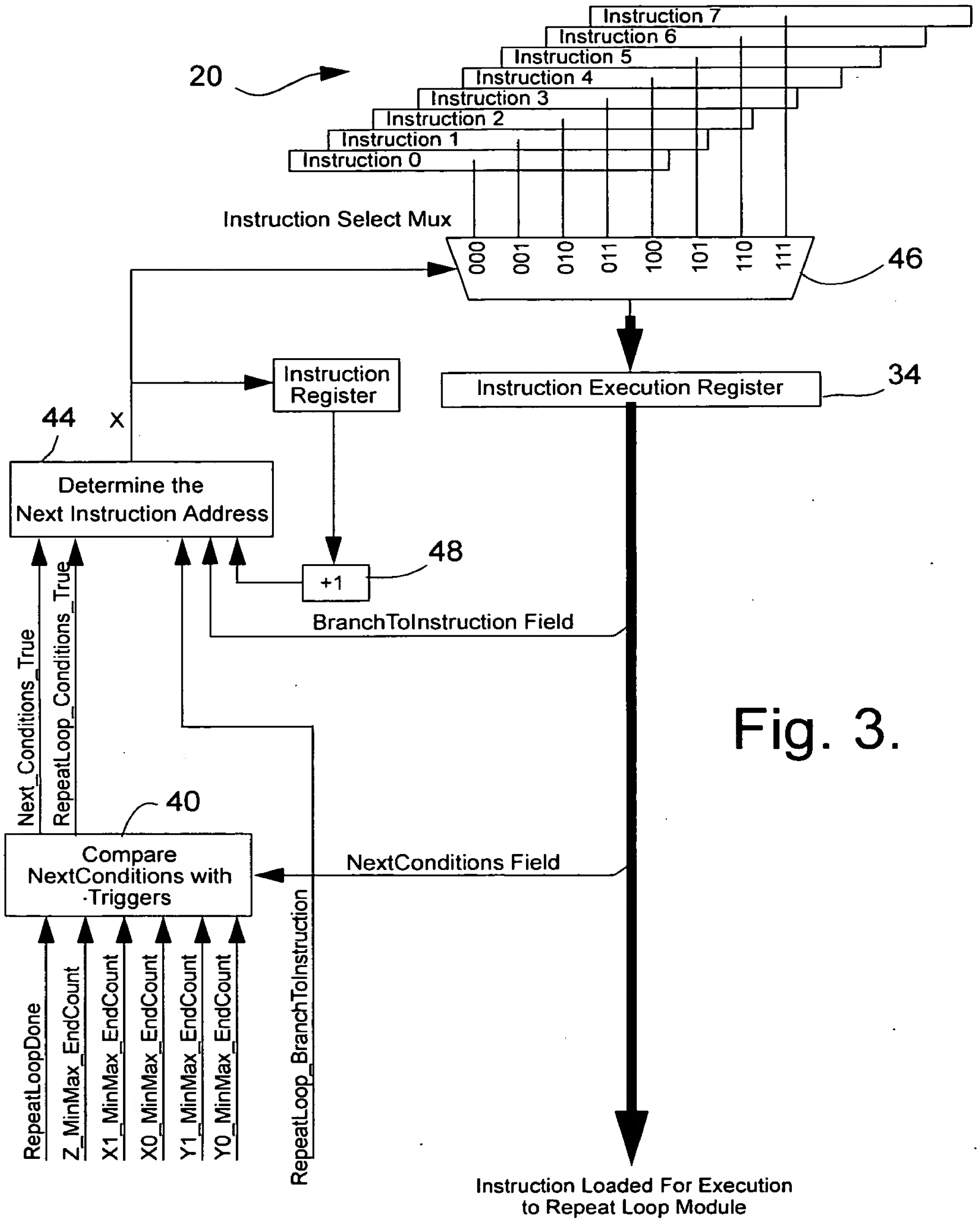

Fig. 3.